MAIL FROM BBN-TENEXA RCVD AT 17-JUL-75 1201-EDT
DATE: 17 JUL 1975 1405-EDT
FROM: BURCHFIEL AT BBN-TENEXA
SUBJECT: PRTN 147 FOLLOWS (ABOUT 5 PAGES)
TO: PACKET-RADIO-GROUP:
CC: STRAZISAR, BEELER

Wax

PRTN 147
17 JULY 1975

ELF MODIFICATIONS AT BBN -- SPRING AND SUMMER 1975

VIRTUAL MEMORY MODIFICATIONS

THE CURRENT VIRTUAL ELF SYSTEM DOES NOT PROVIDE SEVERAL CAPABILITIES DESIRABLE IN A VIRTUAL SYSTEM. IN PARTICULAR, FOR THE PRN STATION THE FOLLOWING CAPABILITIES ARE NEEDED:

1. PAGE FAULTING, A FATAL ERROR ON THE 11/40 SHOULD BE PREVENTED. IF IT IS IMPOSSIBLE TO PREVENT ALL PAGE FAULTS, THE SYSTEM SHOULD LIMIT DAMAGE TO THE FAULTING PROCESS AND PREVENT ANY SYSTEM PROCESS, INCLUDING THE DEBUGGER, FROM FAULTING.
2. THERE SHOULD BE A CAPABILITY FOR DYNAMICALLY DEFINING AND LOADING VIRTUAL ADDRESS SPACES VIA THE DEBUGGER.
3. IN RESPONSE TO EXAMINE AND DEPOSIT COMMANDS, THE DEBUGGER SHOULD BE ABLE TO REFERENCE ANY LOCATION IN THE USER'S ADDRESS SPACE WITHOUT HAVING TO CHECK THE USER'S MAP TO INSURE THAT THE PAGE IS DEFINED.
4. THE VIRTUAL SYSTEM SHOULD BE AS INVISIBLE AS POSSIBLE TO THE USER.

IN THE FOLLOWING, THE MODIFICATIONS ARE PRESENTED AND DISCUSSED IN TERMS OF THESE CAPABILITIES. IT IS THE INTENT TO EXPLAIN WHAT ALTERNATIVES WERE CONSIDERED, WHAT MODIFICATIONS WERE PROPOSED AND HOW THESE MODIFICATIONS ARE HELPING TO ACHIEVE THE ABOVE CAPABILITIES.

THE MOST IMPORTANT CAPABILITY TO CONSIDER IS PREVENTING PAGE FAULTS, BOTH BECAUSE PAGE FAULTS ARE IRRECOVERABLE ON THE 11/40 AND BECAUSE THE MODIFICATIONS SUGGESTED HERE CAN BE USED TO ACHIEVE THE OTHER CAPABILITIES OUTLINED ABOVE. IN PREVENTING PAGE FAULTING, TWO CASES MUST BE CONSIDERED:

1) A PROCESS REFERENCES ITS OWN ADDRESS SPACE, AND
2) A PROCESS REFERENCES AN ADDRESS SPACE OTHER THAN ITS OWN.

THE CHANGES OUTLINED BELOW WILL PROVIDE EACH PROCESS (OR SET OF PROCESSES IN THE SAME ADDRESS SPACE) UP TO 32K OF CORE AS NEEDED. WHEN A PROCESS IS CREATED AND LOADED VIA THE DEBUGGER, THE DEBUG PROCESS WILL MAKE THE APPROPRIATE SYSTEM CALLS TO DEFINE AS MANY PAGES AS ARE NEEDED FOR THE CODE AND DEFINED STORAGE. THE KERNEL WILL ALLOCATE A USER STACK AND WILL DEFINE THE PAGE OR PAGES WHICH THE STACK USES. IN ADDITION, THE PROGRAM MAY, VIA SYSTEM CALLS, DEFINE AND LOAD ADDITIONAL PAGES TO EXPAND ITS STORAGE DYNAMICALLY. HOWEVER, IF THE PROGRAM REFERENCES A LOCATION NOT DEFINED EITHER INITIALLY, OR VIA SYSTEM CALLS WITHIN THE PROGRAM, A PAGE FAULT WILL OCCUR WHICH WILL CAUSE DE-SCHEDULING OF THIS PROCESS. THE PROCESS'S ADDRESS SPACE MAY THEN BE EXAMINED AND MODIFIED VIA THE DEBUGGER, AND THE PROCESS MAY BE RESTARTED (AT SOME RESTART POINT). NO OTHER PROCESS WILL BE AFFECTED BY THE PAGE FAULT.

TO PREVENT PAGE FAULTING WHEN A PROCESS REFERENCES ITS OWN ADDRESS SPACE, THE ENTIRE 8 PAGE ADDRESS SPACE MUST BE DEFINED AND RESIDENT BEFORE THE PROCESS IS RUN. PRESUMABLY THE SOFTWARE NEEDED FOR THIS COULD BE INCLUDED IN THE SCHEDULER. THE BASIC LIMITING FACTOR IN THIS APPROACH IS THE NEED FOR 32K OF CORE FOR EACH VIRTUAL ADDRESS SPACE CREATED. OUR CURRENT 11/40 HAS 48K CORE WHICH ALLOWS ONLY ONE VIRTUAL ADDRESS SPACE -- THE KERNEL'S. (NOTE THAT THE XNCP AND DEBUGGER WILL RESIDE IN THE KERNEL'S ADDRESS SPACE.) ALTHOUGH WE ANTICIPATE EXPANDING THE CORE ON THIS MACHINE, WE DO NOT BELIEVE IT WILL BE POSSIBLE TO

PROVIDE ENOUGH COMPLETE 32K ADDRESS SPACES FOR ALL THE STATION PROCESSES.

IF THE USER REFERENCES AN UNDEFINED LOCATION VIA AN EXAMINE OR DEPOSIT COMMAND TO THE DEBUGGER, THE PAGE WILL BE DEFINED BY MAPPING IT TO A CLEARED FREE PHYSICAL PAGE. IN THIS MANNER, A USER IS PREVENTED FROM CRASHING AN INSTANCE OF THE DEBUG PROCESS BY SPECIFYING AN UNDEFINED ADDRESS. (NOTE THAT THIS IS A CASE OF MOVING DATA BETWEEN ADDRESS SPACES, ONE OF WHICH, THE DEBUG PROCESS, MAY BE IN KERNEL SPACE.)

FINALLY, IF A PROCESS REFERENCES A PAGE IN AN ADDRESS SPACE OTHER THAN ITS OWN, AND THE PAGE IS UNDEFINED, IT WILL BE DEFINED BY MAPPING IT TO A CLEARED FREE PHYSICAL PAGE.

IN SUMMARY, PAGE FAULTS CANNOT BE PREVENTED IN ALL CASES, BECAUSE IT IS IMPOSSIBLE TO GUARANTEE EACH PROCESS A 32K ADDRESS SPACE. PAGE FAULTS OCCUR ONLY WHEN A PROCESS REFERENCES AN UNDEFINED PAGE IN ITS OWN ADDRESS SPACE. WHEN A PAGE FAULT OCCURS, ONLY THE FAULTING PROCESS CRASHES.

THE MODIFICATIONS NEEDED TO MEET THE ABOVE SPECIFICATION ARE AS FOLLOWS:

$PGFLT PAGE FAULT HANDLER

CALLED BY SYSTEM CALLS WHICH REFERENCE ANY ADDRESS SPACE OTHER THAN THE KERNEL'S OR THE ACTIVE PROCESS'S. CHECKS IF PAGE REFERENCED IS RESIDENT AND CAUSES SYSTEM ERROR IF NON-RESIDENT. WILL BE MODIFIED TO CHECK IF PAGE REFERENCED IS DEFINED. UNDEFINED PAGES WILL BE DEFINED BY MAPPING THEM TO CLEARED FREE PHYSICAL PAGES VIA THE $DEFPG SYSTEM CALL. IF $DEFPG RETURNS A SEMAPHORE INDICATING NO FREE CORE, $PGFLT WILL WAIT ON THIS SEMAPHORE; THUS ON RETURN FROM A CALL TO $PGFLT THE PAGE IS ALWAYS RESIDENT.

$DEFPG DEFINE A VIRTUAL PAGE

CURRENTLY DEFINES A VIRTUAL PAGE BY MAPPING IT TO A SPECIFIED PHYSICAL PAGE. WILL BE MODIFIED TO ACCEPT -1 AS A PHYSICAL PAGE INDEX MEANING MAP THE VIRTUAL PAGE TO A FREE PHYSICAL PAGE AND CLEAR THE PHYSICAL PAGE. WILL ALSO PERMIT REMAPPING A PREVIOUSLY DEFINED VIRTUAL PAGE TO A SPECIFIED PHYSICAL PAGE. (NOTE: THESE CHANGES EXCEPT FOR CLEARING NEWLY ASSIGNED PHYSICAL PAGES HAVE RECENTLY BEEN ADDED BY SCRL.)

$VMOV VIRTUAL MOVE BETWEEN TWO ADDRESS SPACES

CURRENTLY ASSUMES IN A MOVE BETWEEN KERNEL SPACE AND THE ACTIVE PROCESS'S SPACE, OR WITHIN THE ACTIVE PROCESS'S SPACE, THAT THE ACTIVE PROCESS'S PAGES ARE RESIDENT. WILL BE MODIFIED TO CALL $PGFLT BEFORE REFERENCING THE ACTIVE PROCESS'S PAGES.

Digitized by the Internet Archive
in 2025 with funding from
Amateur Radio Digital Communications, Grant 151

https://archive.org/details/temporary-note-147

$MTRAP PAGE FAULT PROCESSOR

CALLED AS THE RESULT OF A HARDWARE PAGE FAULT TRAP. CURRENTLY CAUSES A SYSTEM ERROR WHEN RUN ON AN 11/40. WILL BE MODIFIED TO FREEZE (DE-SCHEDULE) THE FAULTING PROCESS.

$DVAS DELETE VIRTUAL ADDRESS SPACE

CURRENTLY RELEASES THE VIRTUAL STORAGE MAP WITHOUT UPDATING THE PHYSICAL STORAGE TABLE (PST). WILL BE MODIFIED TO RELEASE UNSHARED PHYSICAL PAGES BY UPDATING ENTRIES IN THE PST. (ALSO RECENTLY MODIFIED BY SCRL.)

$GSTAK GET A USER STACK

CALLED WHEN A PROCESS IS CREATED TO ALLOCATE VIRTUAL STORAGE IN THE PROCESS'S ADDRESS SPACE FOR A STACK. CURRENTLY DOES NOT GUARANTEE THAT THE PAGE ALLOCATED IS DEFINED. ON AN 11/40, IF THE PAGE IS UNDEFINED, A PAGE FAULT WILL OCCUR WHEN THE PROCESS REFERENCES ITS STACK. WILL BE MODIFIED TO CALL $PGFLT TO GUARANTEE THAT THE PAGE OR PAGES ALLOCATED FOR THE STACK ARE DEFINED.

GIVEN THE ABOVE SET OF MODIFICATIONS, THE DEBUGGER CAN CREATE AND LOAD A VIRTUAL ADDRESS SPACE USING THE FOLLOWING SET OF SYSTEM CALLS:

$CVAS CREATE A VIRTUAL ADDRESS SPACE

ASSIGNS A VIRTUAL STORAGE MAP. A VIRTUAL STORAGE MAP ID (VSMID) IS RETURNED AND IS USED TO CREATE A PROCESS TO RUN IN THIS ADDRESS SPACE. NOTE: IF A USER CALLS $CVAS AND THEN $DVAS, $DVAS TRIES TO RELEASE A SEMAPHORE NEVER DEFINED. THIS WILL BE FIXED BY MODIFYING $CVAS TO CALL $IVAS.

$IVAS INITIALIZE A VIRTUAL ADDRESS SPACE

CLEARS THE ENTRIES IN A VIRTUAL STORAGE MAP, MAKING ALL PAGES UNDEFINED. (CURRENTLY, ON CREATION, THE MAP CONTAINS ENTRIES SET UP BY THE LAST PROCESS TO USE THE MAP.)

$CREAP CREATE A PROCESS

CALLER SPECIFIES A VSMID. A PROCESS IS CREATED TO RUN IN THIS ADDRESS SPACE.

$SIO START I/O

LOAD THE NEW ADDRESS SPACE USING INTER-PROCESS PORTS, INPUTTING DATA TO BE LOADED FROM THE DEBUG PROCESS ON THE PDP-10 AND OUTPUTTING IT TO AN ADDRESS IN THE NEW ADDRESS SPACE. WHEN USING INTER-PROCESS PORTS, THE $SIO CALL USES $VMOV TO ACTUALLY TRANSFER WORDS BETWEEN ADDRESS SPACES. WITH $VMOV MODIFIED AS ABOVE, AS EACH PAGE IN THE NEW ADDRESS SPACE IS REFERENCED, IT WILL BE DEFINED.

$GPCT GET PROCESS CONTROL TABLE WORD

CURRENTLY THERE IS NO ADEQUATE MEANS TO OBTAIN VARIABLES DESCRIBING A PROCESS. ADDITION OF THIS SYSTEM CALL WILL PROVIDE SIMPLE ACCESS TO A PROCESS CONTROL TABLE (PCT) WHERE VARIOUS DATA SPECIFIC TO THE PROCESS IS STORED.

A FURTHER CONSEQUENCE OF THE ABOVE MODIFICATIONS IS THAT THE DEBUGGER MAY REFERENCE ANY LOCATION IN A USER'S ADDRESS SPACE REGARDLESS OF WHETHER OR NOT THE PAGE REFERENCED IS DEFINED. THE DEBUGGER ISSUES AN $SIO CALL ON AN INTER-PROCESS PORT AND SPECIFIES AN ADDRESS IN THE USER'S ADDRESS SPACE. AS EXPLAINED ABOVE, THE $SIO CODE INCLUDES A CALL TO $VMOV WHICH CALLS $PGFLT TO DEFINE ANY UNDEFINED PAGES, THUS PREVENTING PAGE FAULTS.

THE FINAL CAPABILITY TO CONSIDER IS THAT THE VIRTUAL SYSTEM SHOULD BE INVISIBLE TO THE USER. WITH THE ABOVE MODIFICATIONS, THE USER MAY CREATE PROCESSES AND LOAD PROGRAMS VIA THE DEBUGGER IN THE SAME MANNER AS IN A NON-VIRTUAL SYSTEM. THE VIRTUAL SYSTEM PROVIDES THE USER POWERFUL CAPABILITIES IN CREATING AND USING MULTIPLE ADDRESS SPACES. BUT IN THE SIMPLEST CASE OF A SINGLE PROCESS EXECUTING A PROGRAM IN A SEPARATE ADDRESS SPACE, THE DEBUGGER CAN ESTABLISH THE PROPER VIRTUAL ENVIRONMENT THUS MAKING THE VIRTUAL CAPABILITIES COMPLETELY INVISIBLE TO THE USER.

AC BLOCK MODIFICATION

THE INITIAL IMPLEMENTATION OF THE ELF DEBUG PROCESS DID NOT ALWAYS HAVE ACCESS TO A CONSISTENT SET OF "USER" REGISTERS. ("USER" IN QUOTES BECAUSE SOME ELF PROCESSES ACTUALLY RUN IN KERNEL MODE.) THIS WAS THE CASE BECAUSE A COMPLETE CONTEXT SAVE WAS NEVER MADE UNTIL A PROCESS BECAME DESCHEDULED. UNTIL THIS TIME ALL VOLATILE REGISTERS WERE PUSHED ONTO THE KERNEL STACK AS NEEDED. BECAUSE OF THIS THE SET OF REGISTERS WHICH THE PROCESS SAW WHEN RUNNING AT PROCESS LEVEL (I.E., THE OUTERMOST CONTEXT) WERE IN UNDEFINED LOCATIONS ON THE KERNEL STACK, DEPENDING UPON WHICH KERNEL ROUTINES THE PROCESS HAD EXECUTED. THE SET OF REGISTERS MADE ACCESSIBLE BY THE $GPREG (GET PROCESS REGISTER) AND $SPREG (SET PROCESS REGISTER) PRIMLITIVES WERE THOSE OF THE INNERMOST CONTEXT (THE ONLY ONES EXPLICITLY SAVED EN MASSE). THIS OF COURSE IS EXACTLY THE WRONG SET OF REGISTERS TO PRESENT TO A USER WHO IS DEBUGGING A PROCESS!

TO SOLVE THIS PROBLEM A MECHANISM WAS ADDED TO ELF TO SAVE THE CONTEXT OF A PROCESS IN A SPECIAL AREA, CALLED THE User AC Block, OF THE PROCESS CONTROL TABLE (PCT) WHENEVER A PROCESS LEAVES PROCESS LEVEL BY EXECUTING A SYSTEM CALL (EMULATOR TRAP, EMT) OR BREAKPOINT TRAP (BPT) INSTRUCTION, OR BECOMES DESCHEDULED WHEN AN I/O OPERATION COMPLETES. THE AC Block IS ADDRESSED BY $SPREG AND $GPREG, SO THAT THE REGISTERS THE DEBUGGER SEES ARE THE SAME THAT THE PROCESS SEES AT PROCESS LEVEL. THE VALUE OF THE KERNEL STACK POINTER (KSP) IS ALSO SAVED IN A SPECIAL CELL IN THE PCT SO THAT THE KERNEL STACK CAN BE PROPERLY RESET.

WHEN THE CONTEXT IS SAVED THE RETURN PC AND PS ON THE STACK ARE REPLACED WITH A SO-CALLED "FAKE PC AND PS" WHICH POINT AT THE CONTEXT

RESTORE ROUTINE. THIS ALLOWS THE AC Block MECHANISM TO BE ADDED WITHOUT CHANGING ANY OF THE CODE IMPLEMENTING THE SYSTEM CALLS. NOTE THAT REGISTERS ARE NOT RESTORED AFTER SYSTEM CALLS BECAUSE SOME SYSTEM CALLS RETURN VALUES IN THE REGISTERS.

@LOGO